Pointer Handling Questions

Q.1

Write a C program that declares an integer variable 'num' and initializes it with 5. Declare a pointer variable and assign the address of 'num' to it. Use the pointer to increment the value of 'num' by 10. Print the value of 'num' using both the pointer and the variable itself.

Example:

Output: Using variable: 15, Using pointer: 15

[4]

Q.2

Write a C program that reads 5 integers from the user and stores them in an array. Use a pointer to calculate the sum of all the elements in the array. Print the sum.

Example:

Input: 1 2 3 4 5

Output: Sum: 15

[4]

Q.3

Write a C program that defines a function 'swap' to swap the values of two integer variables using pointers. Call this function in the main function to swap two integers entered by the user.

Example:

Input: a = 5, b = 10

Output: After swapping: a = 10, b = 5

[4]

Q.4

Write a C program that declares a 2D array of integers and uses a pointer to find the maximum value in the array. Print the maximum value and its position (row and column) in the array.

Example:

Input: 1 2 3

4 5 6

7 8 9

Output: Maximum Value: 9, Position: (2, 2)

[4]

Q.5

Write a C program that takes two strings as input and uses pointers to concatenate them. Print the concatenated string.

Example:

Input: 'Hello', 'World'

Output: 'HelloWorld'

[4]

Q.6

Write a C program that defines a function to reverse an array using pointers. Take an array of 5 integers from the user, reverse it using the function, and print the reversed array.

Example:

Input: 1 2 3 4 5

Output: Reversed Array: 5 4 3 2 1

[4]

Q.7

Write a C program that dynamically allocates memory for an array of 10 integers using pointers. Take input for the array elements from the user, calculate the average of the elements, and print it. Finally, free the allocated memory.

Example:

Input: 1 2 3 4 5 6 7 8 9 10

Output: Average: 5.5

[4]

Q.8

Write a C program that defines a structure 'Rectangle' with members length and breadth. Write a function to calculate the area of the rectangle using a pointer to the structure. Take input for 3 rectangles and display the rectangle with the largest area.

Example:

Input: Rectangle1: 5, 4

Rectangle2: 6, 3

Rectangle3: 7, 2

Output: Largest Area: 20

[4]

Q.9

Write a C program that declares an array of integers. Use a pointer to find and print all the prime numbers in the array.

Example:

Input: 1 2 3 4 5 6 7 8 9 10

Output: Prime Numbers: 2 3 5 7

[4]

Q.10

Write a C program that defines a function to find the sum of elements in an array using a pointer. Take an array of integers as input from the user. Calculate the sum of elements in the subarray starting from index 2 to index 5 using the function. Print the sum.

Example:

Input: 1 2 3 4 5 6 7

Output: Sum of subarray: 18

[4]

Structure Related Practice Problems

1. **a)** Identify and correct the errors in the following code:

```
struct employee {
   char name[];
   int emp_id;
}
int main() {
   employee e1, e2;
   e1.name = "Alice";
   e1.emp_id = 202;
   struct employee* e_ptr = e2;
   scanf("%s", &e_ptr.name);
   scanf("%d", &e_ptr.emp_id);
}
```

   b) Write a C program to store the following information about books and perform the following operations:

   i. Create a structure named Book with the following members: title (string), author (string), price (float), and year (int).
   ii. Declare an array of size 50 of type Book structures.
   iii. Take inputs (title, author, price, year) from the keyboard and store them in the array.
   iv. Find and display the details of the most expensive book.

2. **a)** Identify and correct the errors in the following code:

```
struct book {
   char title[50];
   int pages;
}
int main() {
   book b1, b2;
   strcpy(b1.title, "C Programming");
   b1.pages = 300;
   book* b_ptr = b2;
   scanf("%s", b_ptr.title);
   scanf("%f", &b_ptr.pages);
   printf("Title: %s, Pages: %f", b1.title, b1.pages);
   printf("Title: %s, Pages: %f", b2.title, b2.pages);
   return 0;
```

}

b) Write a C program to store the following information about students and perform the following operations:

i. Create a structure named Student with the following members: name (string), roll_number (int), marks (float), and grade (char).

ii. Declare an array of size 50 of type Student structures.

iii. Take inputs (name, roll_number, marks) from the keyboard, and calculate the grade of the student based on the marks using the following criteria:

A for marks >= 90
B for marks >= 80 and < 90
C for marks >= 70 and < 80
D for marks >= 60 and < 70
F for marks < 60

iv. Find and display the details of the student with the highest marks.

v. Count and display the number of students who have received a grade of A.

3. **a)** Identify and correct the errors in the following code:

```
struct movie {
char title[100],
int duration,
float rating;
}
int main() {
    struct movie m1, m2;
    m1.title = "Inception";
    m1.duration = 148;
    m1.rating = "8.8";
    struct movie* m_ptr = m2;
    scanf("%s", m_ptr.title);
    scanf("%f", &m_ptr.duration);
    printf("Title: %s, Duration: %f minutes, Rating: %d", m1.title, m1.duration,
m1.rating);
    return 0;
}
```

b) b) Write a C program to store and manipulate information about books using structure pointers:

i. Create a structure named Book with the following members: title (string), author (string), price (float), and year (int).

ii. Declare an array of type Book with a fixed size (e.g., 100).

iii. Take inputs (title, author, price, and year) for each book from the keyboard using structure pointers.

iv. Find and display the details of the book with the highest price using structure pointers.

v. Count and display the total number of books published after the year 2000 using structure pointers.

4. Suppose you're developing a program for the United International University (UIU) Library to manage book borrowing records. Your task is to create a C program that can store, manage, and manipulate the borrowing data of students over the last year. As it is the end of the year, the library is giving a "top borrower award" to the students who borrowed the most books in the past year. Write a C program that will:

   i. Store the following information of a student in a structure: (i) Name (ii) ID (iii) Number of books borrowed, and (iv) An array of book titles borrowed. Use appropriate data types and variable names for all the features.

   ii. Take input for 50 students from the user.

   ii. Calculate the total number of books borrowed by each student.

   iii. To find the top borrower, only consider the students who have borrowed more than 5 books. Among these selected students, the student who has borrowed the most books will win the award. Print the student's name who has won the award.

5. Suppose you're developing a program for the United International University (UIU) Gym to manage student membership and workout sessions. Your task is to create a C program that can store, manage, and manipulate workout data of students over the last semester. As it is the end of the semester, the gym is giving a "most dedicated member award" to the students who attended the gym the most. Write a C program that will:

   i. Store the following information of a student in a structure: (i) Name (ii) Student ID (iii) Number of workout sessions attended, and (iv) An array of durations (in minutes) for each workout session. Use appropriate data types and variable names for all the features.

   ii. Take input for 60 students from the user.

   iii. Calculate the total workout duration for each student.

   iv. To find the most dedicated member, only consider students who have attended more than 20 workout sessions. Among these selected members, the student with the highest total workout duration will win the award. Print the student's name who has won the award.

6. Write a C Program to store the following information about international athletes and find the "Athlete of the Year":

i) Create a structure named Athlete with the following members:

- name (string of length 50)
- country (string of length 50)
- sport (string of length 30)
- events_participated (in an array for the last 30 events)
- medals_won (in an array indicating the count of each type of medal won in the last 30 events, e.g., gold, silver, bronze)
- total_events (int for the last year)
- overall_performance_score (float)

ii) In the main() function,

a. Take input for 100 athletes from the user. DO NOT take the overall performance as input.

b. Additionally, calculate the overall performance of each athlete in the following way:

i. Calculate the overall performance score as the total number of medals won (sum of gold, silver, and bronze) ÷ total events participated.

c. Find and display the information of the "Athlete of the Year" who has the highest overall performance score.

7. Write a C Program to store the following information about library books and manage borrowing records:

i) Create a structure named Book with the following members:

- title (string of length 100)
- author (string of length 50)
- ISBN (string of length 20)
- year_of_publication (int)
- copies_available (int)
- borrowed_count (int)

ii) In the main() function,

a. Take input for 50 books from the user.

b. Implement a function to allow a user to borrow a book. The function should decrease the copies_available by 1 if there are available copies and increase the borrowed_count by 1. If no copies are available, display an appropriate message.

c. Implement a function to allow a user to return a book. The function should increase the copies_available by 1 and decrease the borrowed_count by 1.

d. Provide an option to display all books along with their details, including the number of copies available and the number of times each book has been borrowed.

8. Write a C program for a university to evaluate students' eligibility for scholarships based on their performance. Do the following operations:

   i) Create a structure named Student with the following members:

   - id (int)
   - name (string of length 50)
   - gpa (float)
   - last_5_semesters_scores (int scores[5]) as members.

   ii) Declare an array of size 100 of type Student structures.

   iii) Take inputs for each student from the keyboard.

   iv) Calculate each student's average score over the last 5 semesters. If the average score is more than 75, print a line stating, “Student with id=(print student’s id here) is eligible for scholarship.” Otherwise, print “Student with id=(print student’s id here) is not eligible for scholarship.”

9. Write a C program for a sports academy to manage the training progress of athletes. Do the following operations:

   i) Create a structure named Athlete with the following members:

   - id (int)
   - name (string of length 50)
   - age (int)
   - training_hours (int hours[12]) representing the number of training hours for each month.

   ii) Declare an array of size N of type Athlete structures.

   iii) Take inputs for each athlete from the keyboard, including their monthly training hours for the last 12 months.

   iv) Calculate the total training hours for each athlete over the last year. If the total training hours are more than 600, print a line stating, “Athlete with id=(print athlete’s id here) has met the training requirement.” Otherwise, print “Athlete with id=(print athlete’s id here) has not met the training requirement.”

10. In a simple online bookstore, each book has a title and a price.

(i) Create a structure named Book with the following members:

- title (string of length 100)
- price (float)

(ii) In the main() function, declare an array of Book structures with a fixed size of 100. Take an integer n for the number of books, then populate the details for each book from user input.

(iii) Write a function applyDiscount(Book *b, float discount) that takes a pointer to a Book structure and a discount percentage. The function should decrease the book's price by the discount amount.

(iv) Write another function getMostExpensiveBook(Book *books, int n) that takes an array of Book structures and returns a pointer to the book with the highest price.

(v) In the main function, call applyDiscount for each book after inputting the details, print the updated price for each book, and then call getMostExpensiveBook to find and print the title and price of the most expensive book.

Recursion Code Output Questions

Q.1

Consider the following C code:

```
#include<stdio.h>
int fun(int n) {
    if (n == 0)
        return 1;
    else
        return n * fun(n - 1);
}
void main() {
    int result = fun(4);
    printf("%d", result);
}
```

What will be the output of the program?

Example:

Output: 24

[4]

Q.2

Consider the following C code:

```
#include<stdio.h>
```

```
void fun(int n) {
    if (n < 1)
        return;
    else {
        printf("%d ", n);
        fun(n - 1);
        printf("%d ", n);
    }
}
void main() {
    fun(3);
}
```

What will be the output of the program?

Example:

Output: 3 2 1 1 2 3

[4]

Q.3

Consider the following C code:

```
#include<stdio.h>
int fun(int n) {
    if (n == 1)
        return 0;
    else
```

```
        return 1 + fun(n / 2);
}
void main() {
    printf("%d", fun(16));
}
```

What will be the output of the program?

Example:

Output: 4

[4]

Q.4

Consider the following C code:

```
#include<stdio.h>
int fun(int x, int y) {
    if (y == 0)
        return 0;
    else
        return x + fun(x, y - 1);
}
void main() {
    printf("%d", fun(3, 5));
}
```

What will be the output of the program?

Example:

Output: 15

[4]

Q.5

Consider the following C code:

```
#include<stdio.h>
int fun(int n) {
    if (n == 0)
        return 0;
    else
        return n + fun(n - 1);
}
void main() {
    printf("%d", fun(5));
}
```

What will be the output of the program?

Example:

Output: 15

[4]

Q.6

Consider the following C code:

```
#include<stdio.h>
void fun(int n) {
    if (n == 0)
        return;
    else {
        printf("%d ", n);
        fun(n / 2);
    }
}
void main() {
    fun(10);
}
```

What will be the output of the program?

Example:

Output: 10 5 2 1

[4]

Q.7

Consider the following C code:

```
#include<stdio.h>
int fun(int n) {
    if (n <= 1)
        return n;
    return fun(n - 1) + fun(n - 2);
```

```
}
void main() {
    printf("%d", fun(6));
}
```

What will be the output of the program?

Example:

Output: 8

[4]

Q.8

Consider the following C code:

```
#include<stdio.h>
int fun(int n) {
    if (n > 100)
        return n - 10;
    else
        return fun(fun(n + 11));
}
void main() {
    printf("%d", fun(95));
}
```

What will be the output of the program?

Example:

Output: 91

[4]

Q.9

Consider the following C code:

```
#include<stdio.h>
void fun(int n) {
    if (n == 0)
        return;
    else {
        fun(n - 1);
        printf("%d ", n);
    }
}
void main() {
    fun(4);
}
```

What will be the output of the program?

Example:

Output: 1 2 3 4

[4]

Q.10

Consider the following C code:

```
#include<stdio.h>
void fun(int n) {
    if (n == 1)
        printf("%d ", n);
    else {
        printf("%d ", n);
        fun(n - 1);
        printf("%d ", n);
    }
}
void main() {
    fun(3);
}
```

What will be the output of the program?

Example:

Output: 3 2 1 2 3

[4]

File Handling Questions

Q.1

Write a C program that reads integers from a file called 'input.txt', calculates the sum of these integers, and writes the sum to a file called 'output.txt'. If the file 'input.txt' does not exist, print an appropriate error message.

Example:

Input: 'input.txt' contains: 1 2 3 4 5

Output: 'output.txt' will contain: Sum: 15

[4]

Q.2

Write a C program that reads text from a file named 'data.txt'. The program should count and print the number of vowels and consonants present in the file. If the file does not exist, print an error message.

Example:

Input: 'data.txt' contains: Hello World

Output: Vowels: 3, Consonants: 7

[4]

Q.3

Write a C program that reads a list of student names and their grades from a file 'students.txt' where each line contains a name followed by a grade. The program should print the name and grade of the student with the highest grade.

Example:

Input: 'students.txt' contains:

Alice 85

Bob 90

Charlie 87

Output: Student with highest grade: Bob, Grade: 90

[4]

Q.4

Write a C program that reads a list of integers from 'numbers.txt', sorts them in ascending order, and writes the sorted list to a file named 'sorted_numbers.txt'. If the file 'numbers.txt' does not exist, print an error message.

Example:

Input: 'numbers.txt' contains: 5 2 9 1 5 6

Output: 'sorted_numbers.txt' will contain: 1 2 5 5 6 9

[4]

Q.5

Write a C program that reads a matrix of integers from a file named 'matrix.txt'. The program should then transpose the matrix and write the transposed matrix to a file named 'transposed_matrix.txt'. Assume the first line of the file contains two integers representing the number of rows and columns.

Example:

Input: 'matrix.txt' contains:

2 3

1 2 3

4 5 6

Output: 'transposed_matrix.txt' will contain:

3 2

1 4

2 5

3 6

[4]

Q.6

Write a C program that reads the content of a file named 'text.txt' and replaces every occurrence of the word 'old' with 'new'. The modified content should be written to a file named 'modified_text.txt'. If the file does not exist, print an error message.

Example:

Input: 'text.txt' contains: This is an old book.

Output: 'modified_text.txt' will contain: This is a new book.

[4]

Q.7

Write a C program that reads a file named 'records.txt' containing a list of employees with their names and salaries. The program should calculate and print the average salary. If the file does not exist, print an error message.

Example:

Input: 'records.txt' contains:

John 50000

Jane 60000

Doe 55000

Output: Average Salary: 55000

[4]

Q.8

Write a C program that reads a binary file named 'data.bin' containing integer values. The program should find and print the maximum value present in the file. If the file does not exist, print an error message.

Example:

Input: 'data.bin' contains: [10, 20, 5, 30, 25]

Output: Maximum Value: 30

[4]

Q.9

Write a C program that reads a file 'paragraph.txt' and identifies the longest word in the file. The program should print the longest word and its length. If there are multiple words with the same maximum length, print the first one.

Example:

Input: 'paragraph.txt' contains: The quick brown fox jumped over the lazy dog.

Output: Longest word: 'jumped', Length: 6

[4]

Q.10

Write a C program that reads a file named 'logs.txt' containing timestamps and messages. The program should filter and print only the messages that occurred between two given timestamps. Take the timestamps as input from the user.

Example:

Input: 'logs.txt' contains:

12:00 Started

12:30 Processed

13:00 Finished

Input timestamps: 12:15, 12:45

Output: Processed

[4]